Architecture Definition Document

Scope

Ce document définit l'architecture source (Baseline Architecture), l'architecture cible (Target Architecture) et fournit une analyse de l'écart entre les deux.

Document écrit selon référence document officiel TOGAF.

Goals, objectives, and constraints

Goals

GitmeMoney, banque internationale, s'intéresse de très près au domaine de la cryptomonnaie. Ce nouveau secteur du métier est identifié comme porteur.

Afin de se placer sur ce marché, la direction décide d'investir sur le développement d'un nouveau produit : une marketplace de la cryptomonnaie connectant les sites marchands de cryptomonnaie et les banques.

Objectives

Le produit devra permettre d'atteindre les objectifs fonctionnels suivants :

- s'inscrire comme client ;
- en tant que client identifié, acheter de la cryptomonnaie ;
- se faire référencer comme vendeur de cryptomonnaie ;
- en tant que vendeur référencé, vendre de la cryptomonnaie ;
- permettre de s'interfacer avec l'API de la plateforme pour de la consultation (ainsi les données pourront être intégrées à des applicatifs tiers).

Contraints

La première contrainte est celle de la **sécurité**. Un produit financier de cette envergure doit être sécurisé tant techniquement que fonctionnellement. Il ne laisse aucune place aux failles.

Une seconde contrainte est celle de la **disponibilité**. Il faut assurer un produit accessible 7 jours sur 7 et 24 heures sur 24, quel que soit le nombre d'utilisateurs simultanés.

Enfin, la dernière contrainte est celle de la **gestion des devises**, ce produit ayant pour vocation une portée internationale.

Baseline Architecture

Business Architecture

D'un point de vue métier, GitmeMoney ne possède actuellement aucun produit qui lui soit propre pour la gestion des cryptomonnaies.

Comme de nombreux autres acteurs financiers, il utilise actuellement Kraken qui est intégré à son système d'information comme applicatif métier tiers.

Acteurs externes

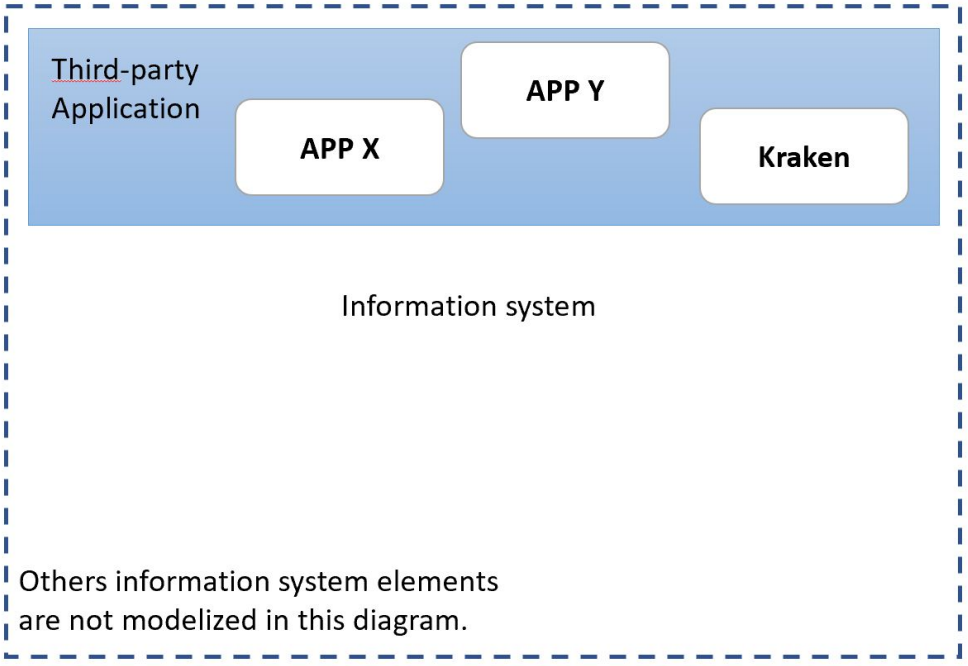
Aucun acteur externe n'est actuellement concerné. Tout est géré en interne.

Acteurs internes

Les employés de la division 'cryptocurrency branch' sont impliqués dans l'utilisation de Kraken.

Actions métiers

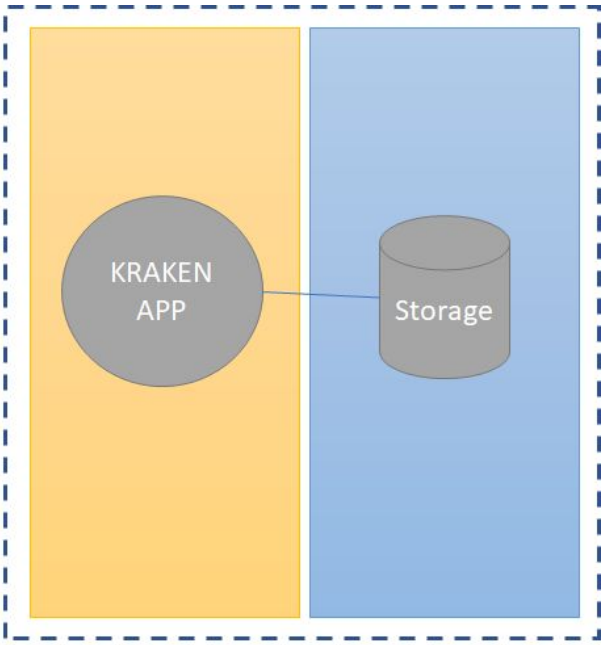
GitmeMoney se place comme utilisateur d'un marketplace (Kraken) pour acheter et vendre sa cryptomonnaie.



Business Architecture models

Data Architecture

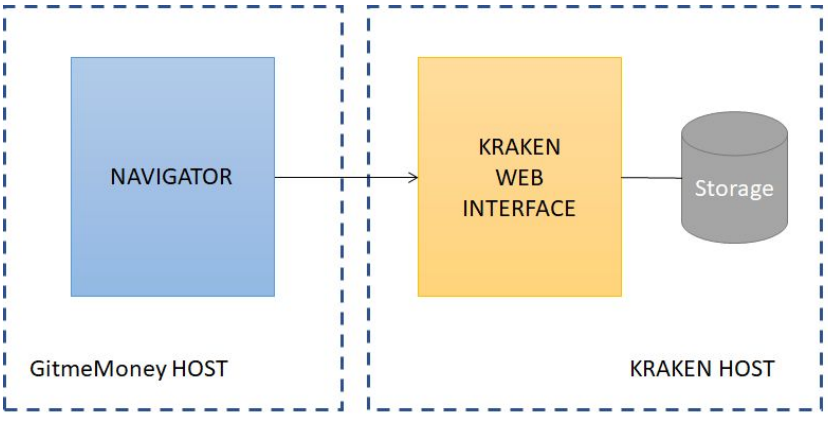
Les données sont actuellement stockées par Kraken, GitmeMoney y accède uniquement par l'interface mise à disposition.



Data Architecture models

Application Architecture

Comme mentionné dans la Business Architecture, GitmeMoney se sert du produit Kraken. En termes d'application, cela signifie utiliser un navigateur web pour accéder aux données à travers l'interface web de Kraken.



Application Architecture models

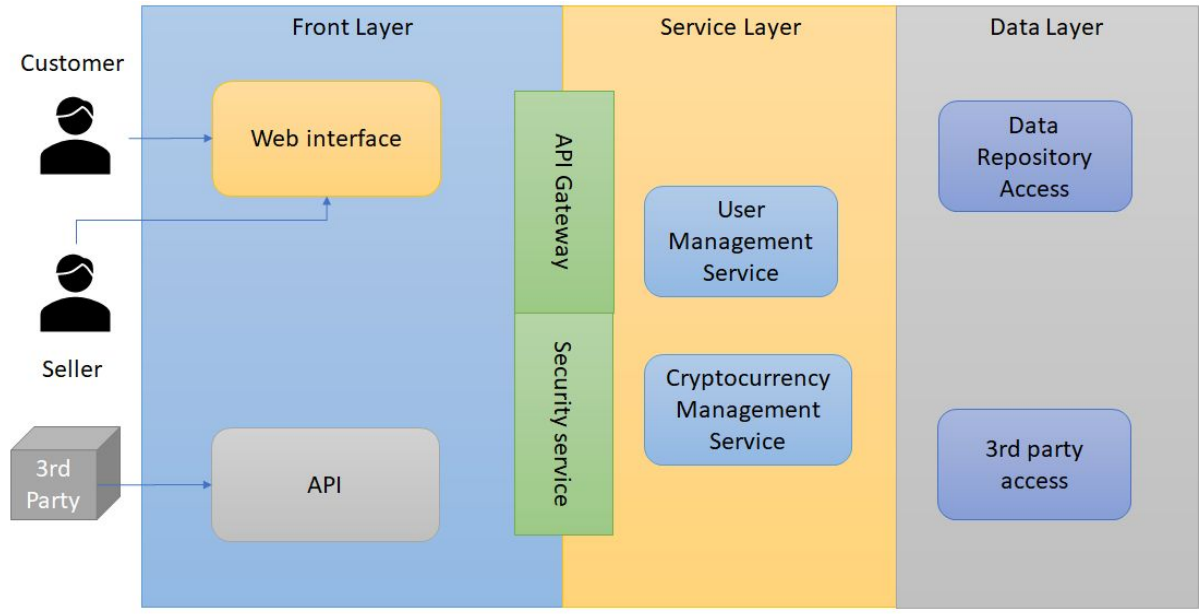
Technology Architecture

GitmeMoney possède un parc informatique respectant de hauts critères de sécurité pour ce qui est des flux entrants et sortants. Aujourd'hui, la plateforme web Kraken est dans la whitelist des sites accessibles par les employés de la division concernée.

Les équipes de développement des produits internes sont principalement axés sur la technologie Java et l'écosystème qui en découle. Selon les produits, il existe des hébergements en local (s'il n'y a pas besoin d'accéder à l'interface en dehors du périmètre interne du réseau) ou des hébergements cloud. Il n'y a pas d'hébergeur Cloud officiel, bien que AWS ait été utilisé à plus haute fréquence.

Target Architecture

L'architecture cible est décrite sur le diagramme suivant :



Target Architecture

Ce diagramme est décrit dans les sous-parties suivantes.

Business Architecture description

Les aspects métiers sont divisés en 3 blocs distincts :

- Backoffice pour les vendeurs.
- Frontoffice pour les acheteurs.
- API sécurisée pour la consultation des données.

Application Architecture description

Front Layer

Permettant aux acteurs d'agir sur le système, il est construit en 2 composants :

1. Interface web (commune au frontoffice et backoffice).
2. API exposée publiquement pour la consultation de données.

Service Layer

Décorrélé du front, il est découpé en 2 blocs métiers :

1. Gestion des utilisateurs.
2. Gestion de la cryptomonnaie (achat/vente).

Les composants 'API Gateway' et 'Security service' sont communs à l'interconnexion du front layer et du service layer.

Data Layer

Cette couche comporte 2 composants :

1. "Data repository access", qui permet d'accéder aux données stockées par GitmeMoney.
2. "3rd party access", qui permet de se connecter à des applications tierces (i.e. se connecter à l'API d'autres produits financiers permettant la récupération d'informations pertinentes à afficher).

Technology Application description

Le produit développé repose sur une stack technique **Java** pour les Service et Data Layer. Le Front Layer doit se reposer sur un framework adaptable à cette stack technique Java.

L'ensemble des composants du produit doit être hébergé sur des serveurs sécurisés (d'un point de vue physique et applicatif).

Gap analysis

Consolidated business

L'application Kraken doit être maintenue tant que le nouveau produit n'est pas stable et que la migration n'a pas été réalisée.

Business gap

GitmeMoney se concentre actuellement sur l'achat et la vente de cryptomonnaie. Ses actions métiers sont toujours valides.

Cependant, GitmeMoney se placera comme une plateforme ouverte pour d'autres acteurs du domaine. Il s'agit donc d'intégrer la gestion de ces utilisateurs.

À noter également que, d'un produit tiers, on passe désormais à un produit propriétaire.

Technical gap

L'écart se résume en l'implémentation d'une application *from scratch* en lieu et place d'un produit tiers intégré dans le système d'information jusqu'à présent.

Ce nouveau produit implique principalement trois nouvelles actions :

1. Implémentation de l'application.
2. Déploiement de l'application.
3. Maintenance de l'application.

Pour chacune de ces phases, des réponses techniques seront fournies en suivant les principes de l'architecture definition document.

Road Map

L'Architecture Roadmap représente le plan architectural pour atteindre la cible définie dans le présent document.